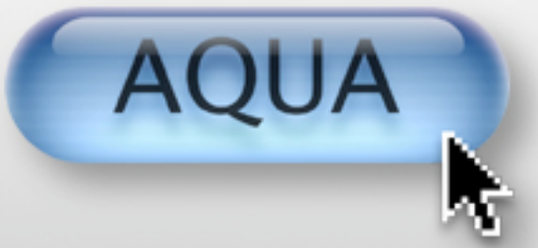
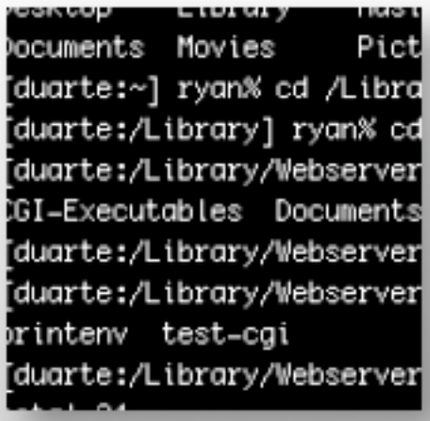
Mac OS X Speech Technologies

Session 402





Mac OS X Speech Technologies

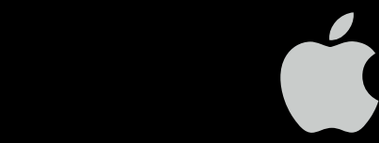
Kim Silverman, PhD
Principal Research Scientist
Manager, Spoken Language Technologies
kimsilv@apple.com

Demo

Speech Features in Panther

Why Adopt Speech?

- Get beyond the limits of a graphical UI
  - Screen real estate
  - What happens when the user isn't attending?
  - Add an extra modality: give the user more choices
- More natural, approachable
  - Some kinds of information best visually, some not
  - Speech is easier in "eyes-busy, hands-busy"
- Move out of the 1980's
  - Beyond "Beep" and "Grunt"



Speech Recognition

- Built-in speaker independent recognition engine
- Works with a far-field microphone
  - Available on lots of Macintosh hardware
  - Robust against background noise
  - Automatically adapts to the acoustic conditions
- Large vocabulary
  - >121,000 words
- Supports continuous speech input
- Driven by a finite state grammar

Why Speech Recognition?

- Speech is a very natural form of communication
- Move beyond limits of “Point and Click”
  - You can’t click on what you can’t point to
- Conversation is an appropriate modality for delegating tasks
- Accessibility!!!

Speech Synthesis

- Unrestricted conversion of text into speech
- Simple, robust, well-documented API
- Range of different voices
- Range of speaking rates
- Ongoing improvement of speech naturalness and intelligibility

When Speech Synthesis?

- Notifications
- Proof reading
- Games
- Education/Literacy
- Web browsing
- Form filling
- Eyes busy, hands busy
- Delegation

Why Speech Synthesis?

- Sometimes recording a voice talent is impractical
- Consistency
- Save space
- Save costs
- Flexible
- Intonation and timing can be controlled
- Lip-synchronization is free

World Book Demo

Jack Minsky
Software MacKiev

Now it’s your turn…

Customizing Speech Synthesis

Speech Synthesis at Its Best

Customize!!!

- Your app knows a lot more about how things should be spoken than the TTS engine does
- Filter the text (AAPL ⇨ “Apple Computer”)
- Customize the pronunciations
- Customize the intonation

Customizing the Pronunciation

- Use funny spelling?
  - Not recommended!
    - Pronunciation of nonstandard spellings can change
  - Unreliable and imprecise
- Use phoneme input instead:
  - My name is
    [[inpt PHON]] _m1AA=tIY=AXs [[inpt TEXT]]

Customizing the Intonation

- What is the problem?
  - Words alone aren't enough
    - "John only introduced Mary to Bill"
- Solution
  - Automatically determining the correct intonation is difficult in the general case
  - Developers often can employ domain knowledge to do much better

Your first flight is with Alaskan Airlines flight 2762 departing from San Jose on Monday May 24th at 6:10 pm landing in San Francisco at 7:10 pm.
Thank you for choosing TTS Travel.

[[rate 20]]

[[pbas +1; pmod +1]] Your [[emph +]] first [[emph -]] flight: is with Alaskan Airlines: “[[emph -]] flight 2762”. [[pbas -1; pmod -1; emph -]] Departing from San Jose, on Monday May twenty fourth: at 6:10 pm, [[slnc 500; emph -]] landing in San Francisco: at [[emph +]] seven [[emph -]] ten [[emph -]] pm!

[[slnc 600]] [[pbas +1; pmod +1]]

Thank you: for [[emph -]] choosing TTS travel.

[[rate -20]]

Your first [[emph -]] flight: is with Alaskan Airlines: “ flight 2762”.

Departing from San Jose, on Monday May twenty fourth: at 6:10 pm, [[slnc 500; emph -]] landing in San Francisco: at [[emph +]] seven ten pm!

Thank you: for [[emph -]] choosing TTS travel.

1. Let the User Catch Up

- Add pauses at major “sense units” or boundaries of internal sentence structure
  - Liberally sprinkle punctuation wherever appropriate
  - Break up larger sentences into smaller ones
  - Lengthen more major pauses by adding extra silence
    - [[slnc 500]]
- Adjust the speaking rate to be appropriate for the purpose of the speech

Departing from San Jose

on Monday May 24th at 6:10 pm

landing in San Francisco at 7:10 pm

[[rate -5]]

Departing from San Jose

on Monday May 24th at 6:10 pm

landing in San Francisco at 7:10 pm

[[rate -5]]
Departing from San Jose:
on Monday May 24th at 6:10 pm,
[[slnc 400]]
landing in San Francisco: at 7:10 pm.

2. Familiar Things Go in the Background

- De-emphasize repeated words

…departing six ten…
…landing seven [[emph -]] ten….

  - Now **automatic** if in the same paragraph

- De-emphasize items inferable from your overall application scenario

Your first [[emph -]] flight is…

3. Liven It Up!

…landing in San Francisco at 7:10 pm!

4. Focus the User's Attention

- Use **[[emph +]]** on important or new information
- Pause with a colon just before the most important item

...at [[emph +]] seven [[emph -]] ten....

5. Use Paragraph Intonation

- Group sentences together into “Intonational paragraphs”, based on topic structure
- Mark them by inserting a blank line
- Raise pitch range for first sentence, then reset

  [[pbas +1; pmod +1]] Your first flight…

  [[pbas -1; pmod -1]] Departing from…
- Put an additional pause between paragraphs

  …at 7:10 pm.

  [[slnc 600; pbas +1; pmod +1]] Thankyou for…

5. Use Paragraph Intonation

- Group sentences together into “Intonational paragraphs”, based on topic structure
- Mark them by inserting a blank line

Your first flight is with Alaskan Airlines flight 2762 departing from San Jose on Monday May 24th at 6:10 pm landing in San Francisco at 7:10 pm.

Thank you for choosing TTS Travel.

Your first flight is with Alaskan Airlines flight 2762 departing from San Jose on Monday May 24th at 6:10 pm landing in San Francisco at 7:10 pm.

Thank you for choosing TTS Travel.

Text to Speech Summary

- Customize pronunciation
  - Use phonemes for mispronounced words
- Customize intonation
  - Let user catch up
  - De-emphasize familiar things
  - Liven it up!
  - Focus the listener's attention on the important elements
  - Use Paragraph Intonation
- Give user a better speech experience

Repeat-After-Me Demo

Kim

Cocoa Classes

Simple things should be simple.
Complex things should be possible.

...Alan Kay

Cocoa Classes

Kevin Aitken

NSSpeechRecognizer

- Designed for fast, easy implementation of speech recognition in your Cocoa application
- No need to understand hierarchical language models just to get started
- Supports dynamic command lists that can change on-the-fly

NSSpeechRecognizer

Getting started

- Create an instance of NSSpeechRecognizer

```
NSSpeechRecognizer * recognizer =
                                [NSSpeechRecognizer new];

[recognizer setDelegate:self];

[recognizer setCommands:[NSArray arrayWithObjects:
                @"North", @"South", @"East", @"West"]];

[recognizer startListening];
```

NSSpeechRecognizer

Getting started

- Set the delegate

```
NSSpeechRecognizer * recognizer =
                                [NSSpeechRecognizer new];

[recognizer setDelegate:self];

[recognizer setCommands:[NSArray arrayWithObjects:
                @"North", @"South", @"East", @"West"]];

[recognizer startListening];
```

NSSpeechRecognizer

Getting started

- Set the commands

```
NSSpeechRecognizer * recognizer =
                                [NSSpeechRecognizer new];

[recognizer setDelegate:self];

[recognizer setCommands:[NSArray arrayWithObjects:
                @"North", @"South", @"East", @"West"]];

[recognizer startListening];
```

NSSpeechRecognizer

Getting started

- Begin listening

```
NSSpeechRecognizer * recognizer =
                                [NSSpeechRecognizer new];

[recognizer setDelegate:self];

[recognizer setCommands:[NSArray arrayWithObjects:
                    @"North", @"South", @"East", @"West"]];

[recognizer startListening];
```

NSSpeechRecognizer

Handling the recognition message

- Your delegate receives a message after each recognition

```
- (void)speechRecognizer:(NSSpeechRecognizer *)sender
                 didRecognizeCommand:(id)command {
  if ([command isEqualToString:@"North"])
    [self moveNorth];
  else if ([command isEqualToString:@"South"])
    [self moveSouth];
  else if ([command isEqualToString:@"East"])
    [self moveEast];
  else if ([command isEqualToString:@"West"])
    [self moveWest];
}
```

NSSpeechRecognizer

Handling the recognition message

- Your delegate receives a message after each recognition

```
- (void)speechRecognizer:(NSSpeechRecognizer *)sender
                 didRecognizeCommand:(id)command {
  if ([command isEqualToString:@"North"])
    [self moveNorth];
  else if ([command isEqualToString:@"South"])
    [self moveSouth];
  else if ([command isEqualToString:@"East"])
    [self moveEast];
  else if ([command isEqualToString:@"West"])
    [self moveWest];
}
```

NSSpeechSynthesizer

- Asynchronous speaking to default sound output device or AIFF file
- Notification of speech events:
  - Speaking completed
  - Phoneme and word
- Access to list of available voices and their properties
  - Name, age, gender
- Integration with NSSpeechRecognizer to create spoken interactions

NSSpeechSynthesizer

Getting started

- Create an instance of NSSpeechSynthesizer

```
NSSpeechSynthesizer * synthesizer =
                                [NSSpeechSynthesizer new];

[synthesizer setDelegate:self];

[synthesizer startSpeakingString:@"Hello"];
```

NSSpeechSynthesizer

Getting started

- Set the delegate

```
NSSpeechSynthesizer * synthesizer =
                                [NSSpeechSynthesizer new];

[synthesizer setDelegate:self];

[synthesizer startSpeakingString:@"Hello"];
```

NSSpeechSynthesizer

Getting started

- Begin speaking through the default output device

```
NSSpeechSynthesizer * synthesizer =
                                [NSSpeechSynthesizer new];

[synthesizer setDelegate:self];

[synthesizer startSpeakingString:@"Hello"];
```

NSSpeechSynthesizer

Getting started

- Or, begin saving the synthesized speech to a file

```
NSSpeechSynthesizer * synthesizer =
                                [NSSpeechSynthesizer new];

[synthesizer setDelegate:self];

[synthesizer startSpeakingString:@"Hello"
        toURL:[NSURL fileURLWithPath:@"~/Hello.aiff"]];
```

NSSpeechSynthesizer

Handling delegate messages

- Message when speaking is finished

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
        didFinishSpeaking:(BOOL)finishedSpeaking {

    // Speaking has finished, update the user interface.

}
```

NSSpeechSynthesizer

Handling delegate messages

- Message when a word is about to be spoken

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
          willSpeakWord:(NSRange)characterRange
               ofString:(NSString *)string {

  // This word is about to be spoken,
  // highlight the word.

}
```

NSSpeechSynthesizer

Handling delegate messages

- Message when a phoneme is about to be spoken

```
- (void)speechSynthesizer:(NSSpeechSynthesizer *)sender
        willSpeakPhoneme:(short)phonemeOpcode {

    // This phoneme is about to be spoken,
    // animate the mouth.

}
```

Demo

Kevin Aitken

Adopting Speech

Adopting Speech

Test with speech enabled

- Start with the Speech preference panel
- Turn on Speakable Items
- Turn on Talking Alerts and Speak Text Under Mouse
- Make sure your controls are spoken and recognized

Adopting Speech

Example of confusing controls

Best

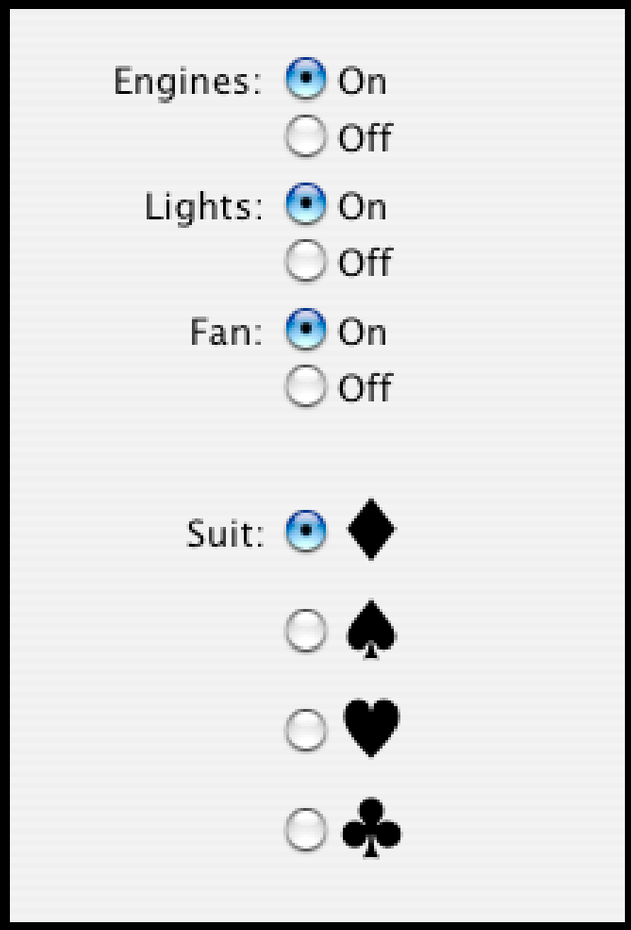
- Provide Spoken Interactions
- Tune Spoken Text
- Offer Spoken Notification

Better

- Document Speech Features
- Customize Spoken Text
- Read Back Information
- Include Spoken Commands

Good

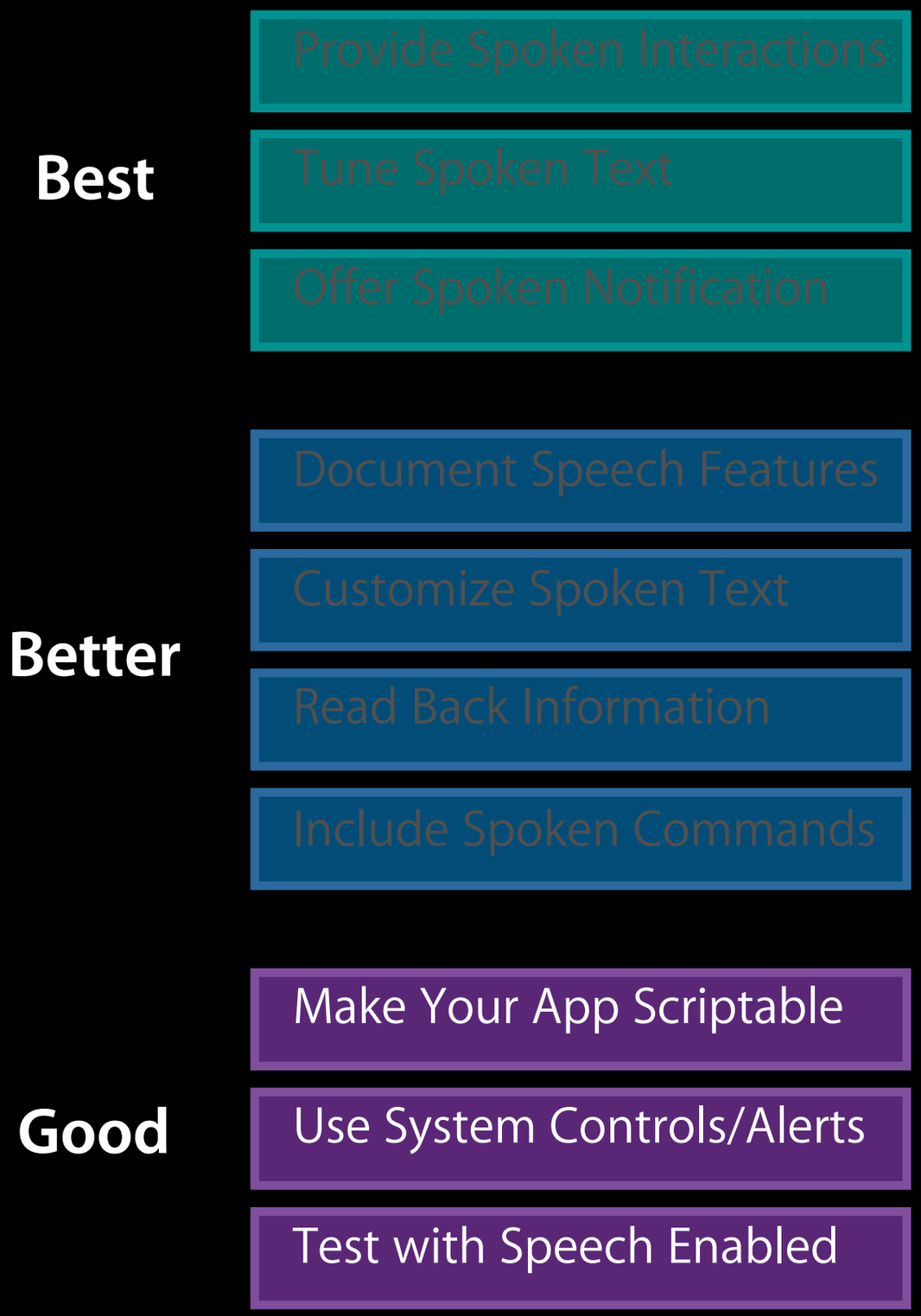
- Make Your App Scriptable
- Use System Controls/Alerts
- Test with Speech Enabled



Adopting Speech

Use system controls and standard alert routines

- Only standard alerts are spoken by Talking Alerts
- If using custom controls follow accessibility documentation under:
  - Carbon User Interface Toolbox
  - Cocoa User Interface Elements



Adopting Speech

Make your application scriptable

- Allows you to easily create spoken commands to include with your application
- Also, enables users to create their own spoken commands:
  - *“remove redeye and print”*

Best

Provide Spoken Interactions

Tune Spoken Text

Offer Spoken Notification

Better

Document Speech Features

Customize Spoken Text

Read Back Information

Include Spoken Commands

Good

Make Your App Scriptable

Use System Controls/Alerts

Test with Speech Enabled

Adopting Speech

Include spoken commands

- Embed items in your application bundle to support Speakable Items
- Or, use new Cocoa classes to implement spoken commands
- Example:
  - Chess: *“Pawn d2 to d4”*

Best

Provide Spoken Interactions

Tune Spoken Text

Offer Spoken Notification

Better

Document Speech Features

Customize Spoken Text

Read Back Information

Include Spoken Commands

Good

Make Your App Scriptable

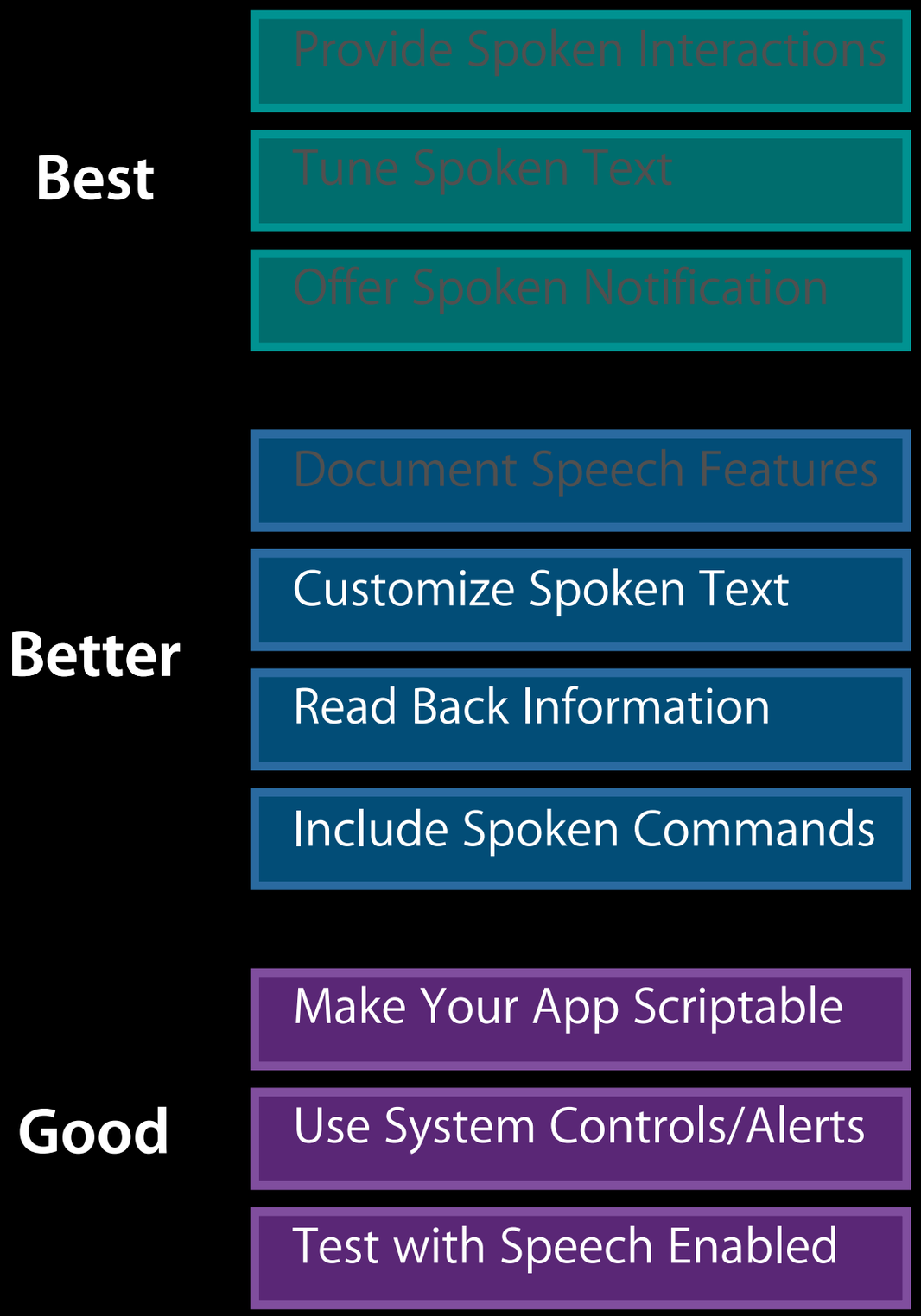
Use System Controls/Alerts

Test with Speech Enabled

Adopting Speech

Read back information

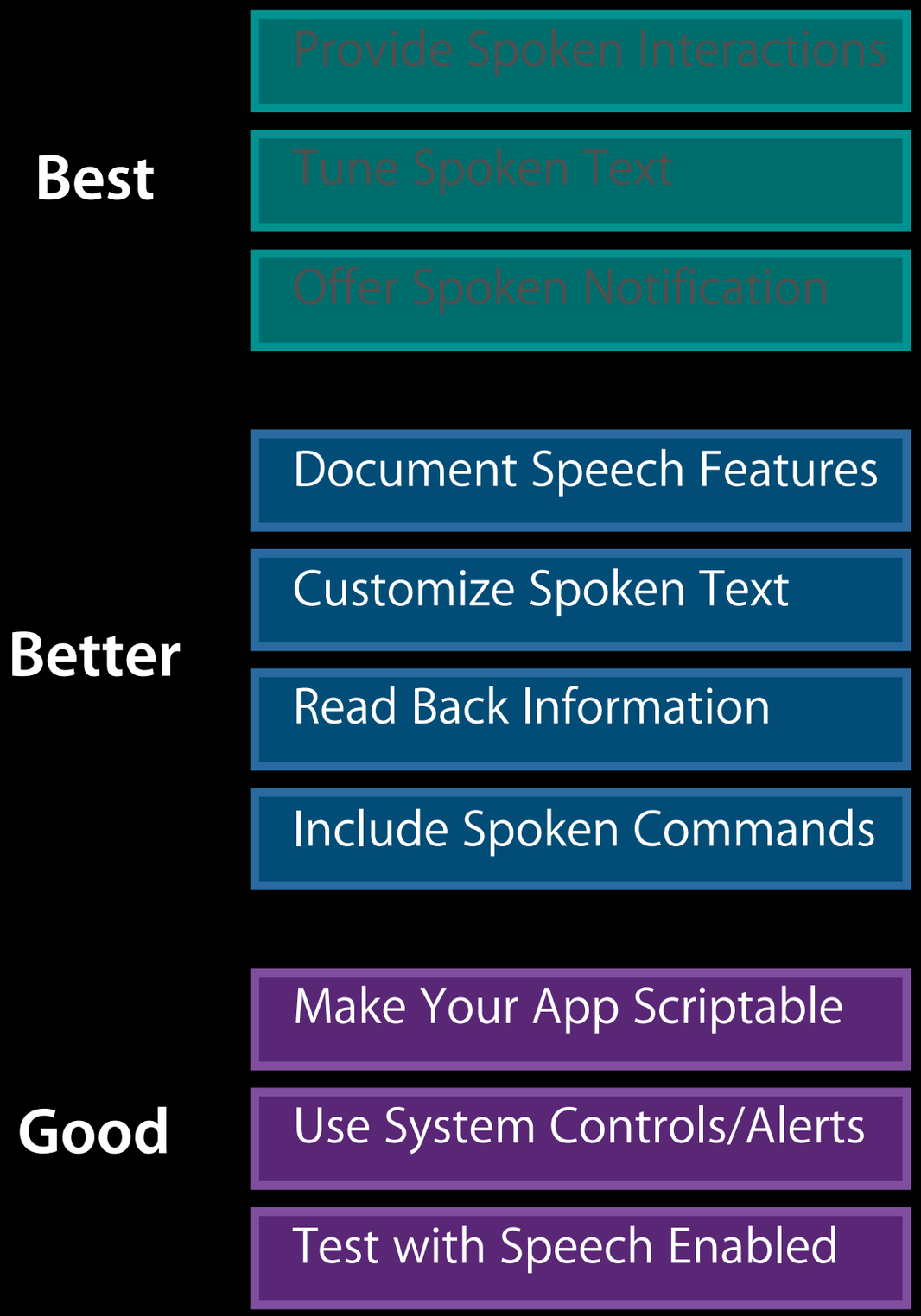
- Offer users the choice to verify on-screen information using audible text
- Examples:
  - Calculator speaks total
  - Spreadsheet program speaks selected cell values
  - Educational program speaks student input



Adopting Speech

Customize spoken text

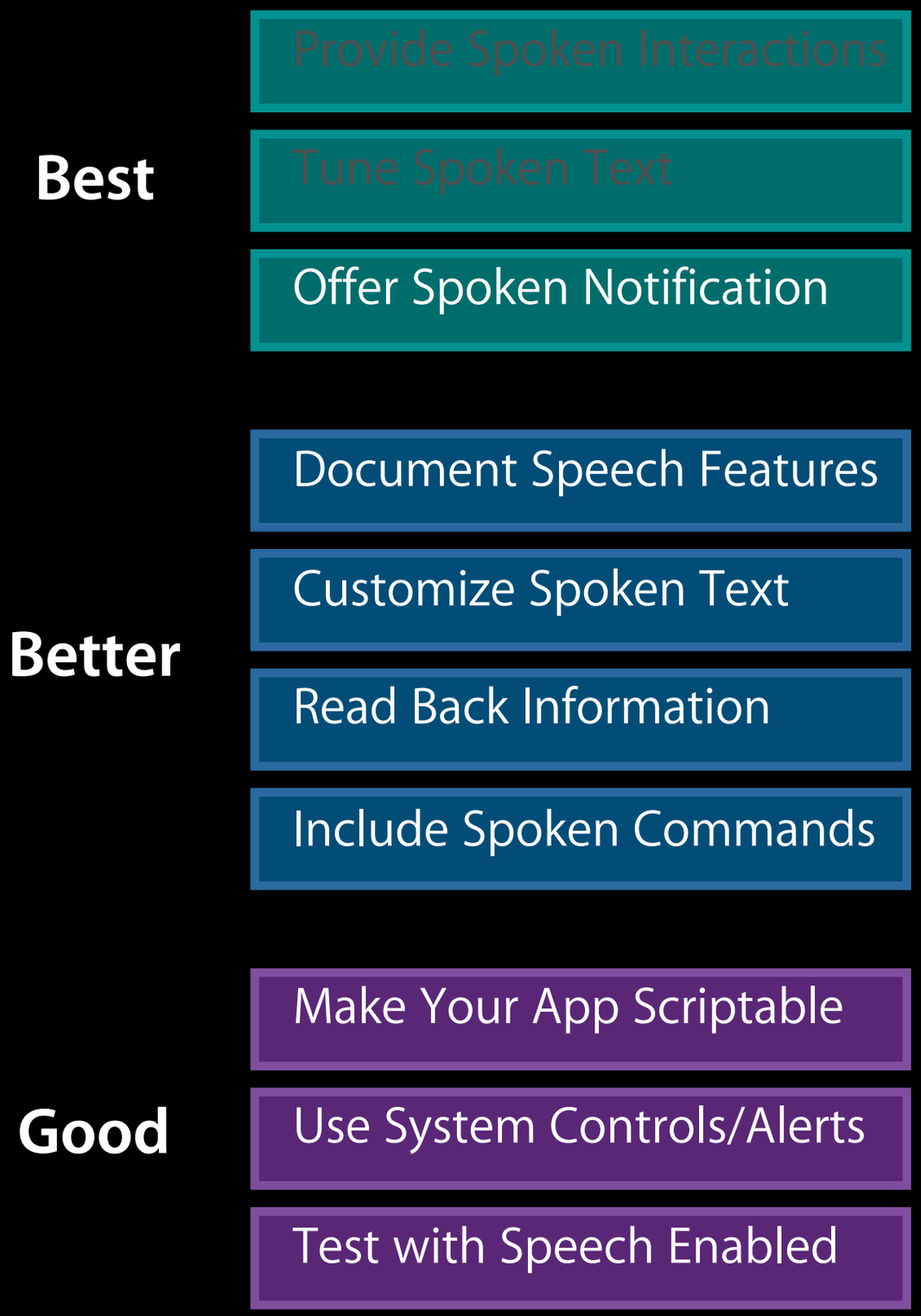
- Customize pronunciation with punctuation and phonetic spelling
  - e.g., *"Linus"*
- Customize intonation using embedded commands
  - e.g., *"Don't even *think* about it!"*



Adopting Speech

Document speech capabilities

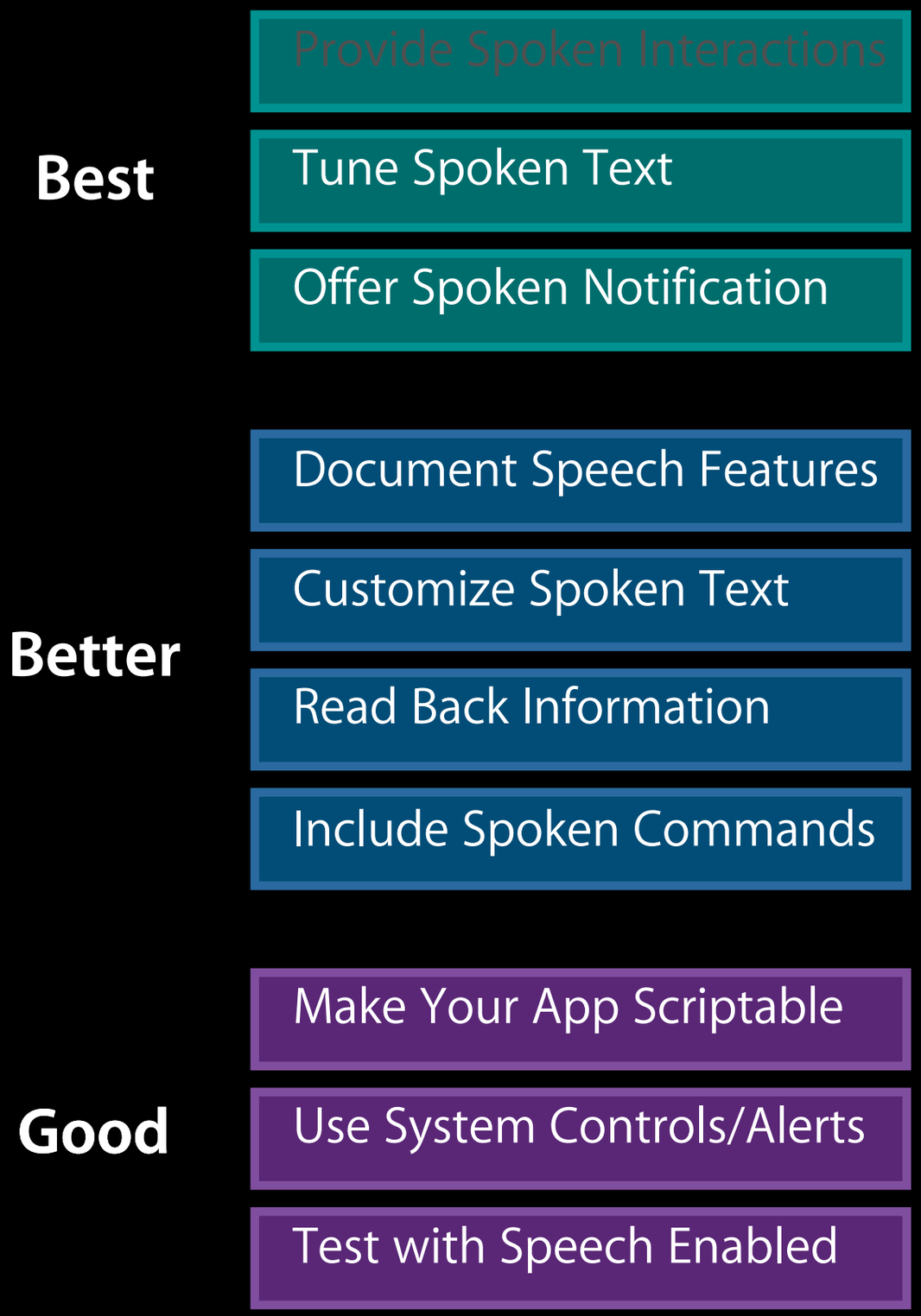
- Describe your application's speech features:
  - In your documentation
  - In your Help files
  - On your box
  - In the user interface
- Point users to Apple's Speech Tips in Mac Help



Adopting Speech

Offer spoken notification

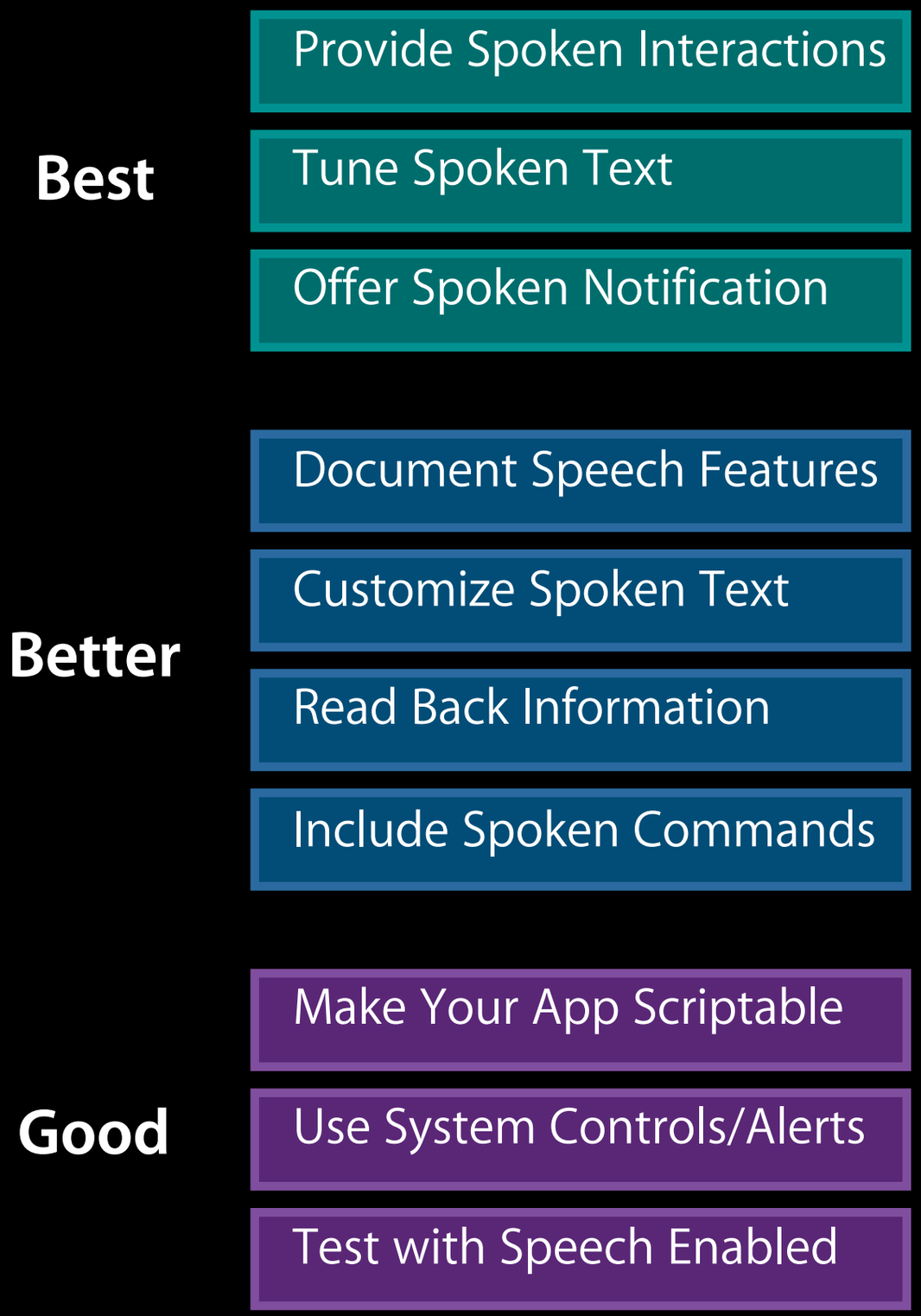
- Provide speech as another notification option
- Examples:
  - iChat: *"Your buddy Carl is available"*
  - Email program: *"New mail from Judy has arrived"*
  - Calendar program: *"You have a meeting with John in 5 minutes"*



Adopting Speech

Tune spoken text

- Use RepeatAfterMe to add the desired emotion to synthesized phrases



Adopting Speech

Provide spoken interactions

- Lead users through tasks using speech
- Example:
  - User: *"Play some music"*
  - App: *"Which artist?"*
  - User: *"Sting"*
  - App: *"Which song?"*
  - User: *"Fields of Gold"*

Wrap Up

Issues

- Educating users how to speak
  - No pauses, don’t shout, …
  - Refer to “Helpful Tips” in Speech preference panel
- Background noise
  - Head-mounted microphones
- Non-native speakers
- Localization

Technology Preview:
New Speech Synthesis Voice Technology

Work in Progress

Formant Synthesis

Diphone Concatenation

Polyphone Viterbi Search

Diphone vs. Polyphone

- Diphone Synthesis:
  - Very short units
  - One choice per unit => one way to synthesize anything
  - Difficult to avoid discontinuities
  - Cannot capture natural co-articulation
- Polyphone Synthesis:
  - Long units: syllables, words, phrases…
  - Exponentially many ways to synthesize anything
  - Captures natural co-articulation

Summary

- Repeat After Me
- New Cocoa Classes
- Guidelines for good, better, and best speech adoption
- Reminder: the underlying technology keeps improving

Roadmap

400 Introduction to Developing Applications with Cocoa	Nob Hill Mon, 5:00pm
401 AppleScript Update	Russian Hill Mon, 5:00pm
421 Mac OS X Accessibility	The Presidio Fri, 10:30am

Who to Contact

John Geleynse
Manager, software Evangelism
geleynse@apple.com

http://developer.apple.com/wwdc2003/urls.html

Resources for Adopting Speech

Speech developers web page
http://developer.apple.com/ue/speech/

Panther Developer CD
/Developer/Examples/Speech/
/Developer/Applications/Extras/Speech/

Speech mailing list
Join at: http://developer.apple.com/ue/speech/

WWDC 2003

Worldwide Developers Conference